GPS spoofing using software defined radio

G. Goavec-Merou[1], J.-M Friedt[1], F. Meyer[2]
[1] FEMTO-ST time & frequency, Besançon ; [2] OSU Theta, Observatoire de Besançon, 13 janvier 2019

The GPS navigation system is before all a time dissemination system used as reference for many applications requiring synchronizing clocks located at spatially distant sites. We demonstrate here how a software defined radio implementation of the sentences emitted by the satellites allows spoofing the position and timing (1 PPS output) of a receiver.

1 Introduction

Navstar, which has now become GPS, is a geolocation system based on trilaterating signals emitted by a satellite constellation. Designed for military purposes, the civilian segment exhibits no protection against attacks. However, such attacks required until recently hardware that was hardly accessible to the general public. This situation is quickly evolving with the availability of software defined radio emitters.

Since the SA (*Selective Availability*) degraded resolution mode has been deactivated in May 2000 [1, 2], GPS has slowly become ubiquitous in most daily activities, if only through our obsession to analyze geolocated information on our smart phones. A British study [3] estimates at 5 billion pounds the losses associated with GPS jamming for 5 days, a trivial task with hardly any technical challenge but emphasizing how ubiquitous satellite navigation systems have become in the critical infrastructures of a country (think of aerial navigation, synchronizing clocks and train transportations, parcel delivery ...). Much more worrisome, we consider here GPS spoofing [4, 5] : while jamming only requires an emitter powerful enough and is immediately detected through the loss of services, spoofing is more subtle since it introduces an erroneous information to the user who believes to have received a valid information, and hence does not realize the attack is being performed [6, 7].

Our purpose is at first to summarize the broad operating principles of GPS [8] : we shall insist on the fact that positioning is based on precise time transfer to allow for trilateration. We then demonstrate the attack on various receivers ranging from smart phones to general public GPS receivers such as those provided by U-Blox (also used in apparatus such as the DJI drones – we leave to the reader's imagination the impact of the attack). Some trivial countermeasures reduce the range and impact of the attack, but do not prevent it, and we shall see that even GPS receivers integrated in vehicles are spoofed after some care is taken on the quality of the generated signal. We conclude with a few countermeasure strategy considerations.

2 GPS basics

GPS, as other satellite navigation systems (Russian GLONASS, European Galileo – more generally Global Navigation Satellite Systems (GNSS)), is made of a constellation of satellites orbiting at an altitude of about twenty thousand kilometers above the surface of the Earth. Celestial mechanics – Kepler's laws – define a few properties on the orbits that will be at the core of our approach to prevent spoofing attacks if appropriate hardware is designed accordingly. Most important, a first parameter we introduce at the beginning of this discussion is the Doppler shift induced by the motion of the satellite. Based on the symbol definitions on Fig. 1, we observe that when a satellite rises above the horizon HH', the angle ϑ is given by $\sin(\vartheta) = R/(R+r)$ with $R = 6400$ km the Earth radius and $r = 20000$ km the satellite altitude along its orbit. Thus, the projection of the tangential component of the velocity vector v is $v_{\parallel} = |\vec{v}|\sin(\vartheta) = v \cdot R/(R+r)$. Considering the third Kepler law stating that the ratio of the square of the period to the cube of the orbit radius is constant, and knowing that geostationary satellites, hence with a period of 24 h, are located at an altitude of 36000 km, we deduce a GPS satellite period of

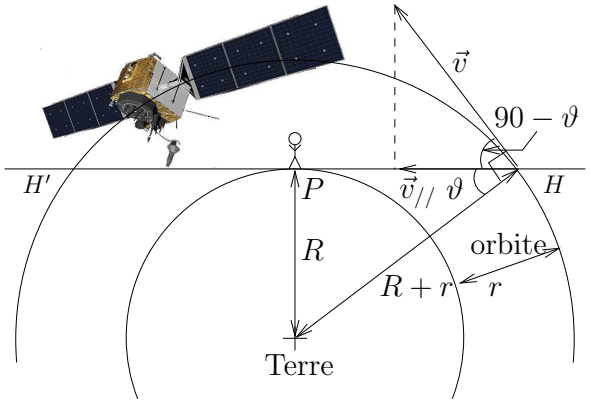


Figure 1: Sketch of a satellite orbit and definition of symbols used in the text.

$T = 12$ h. Based on this period and the distance traveled along the orbit, we deduce a tangential velocity of $2\pi(R+r)/T \simeq 13800$ km/h=3840 m/s. We deduce a maximum radial velocity when the satellite is located in H or H' of $|\vec{v}| = 3840 \times 6400/26400 = 930$ m/s and hence a maximum Doppler shift δf of $\delta f = f_0 \cdot v/c$ with $f_0 = 1575.42$ MHz the carrier frequency and $c = 3 \cdot 10^8$ m/s the velocity of light : $|\delta f| < 4,9$ kHz. This threshold on the **Doppler shift is dictated by celestial mechanics and can never be violated** : we will see that it brings a first protection against GPS signal spoofing.

The information transmitted over the carrier is encoded twice : on the one hand a fast message (1 Mb/s) encodes the identifier (number) of the satellite transmitting the information, and on the other hand the low-datarate (50 bits/s) navigation message sent by each satellite overlaps this code. These various encoding schemes were described in detail in [8], in which we stopped the investigation when navigation message bits had been recovered, without exploiting the message content. The whole challenge of GPS spoofing is to thoroughly rebuild each sentence meeting the requirements of physical parameters of the transmission to get even the most picky receivers to believe the signal they receive is coming from space. The various sentences of the navigation message are described in great detail in [9] : we will obviously not be able in these few pages to repeat the content of the more than 600 pages of these two books whose understanding is mandatory. In particular, these documents explain how to convert satellite orbital parameters (transmitted in the navigation messages) and the transmission date to *pseudo-ranges* including the position of the ground-based receiver. The *pseudo-range* is the core element for positioning the user close to Earth, and the raw information processed by the ground receiver for positioning by trilaterating. These pseudo-ranges are provided as raw data in RINEX (Receiver Independent EXchange Format) files, a standardized format [10] for sharing information between GNSS receivers (Fig. 2).

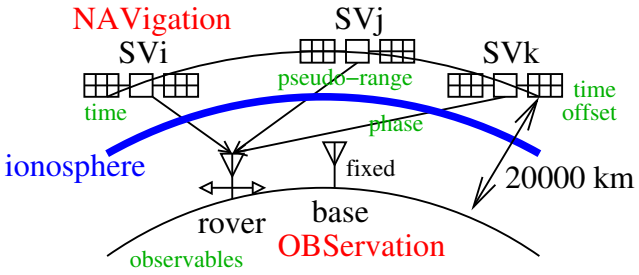


Figure 2: Spatial segment of GPS with the space vehicles (SV), sorted by their pseudo-random code, described in the navigation ephemeris files, and the ground segment described by RINEX observation files.

[10] defines a **pseudo-range** as "*The pseudo-range (PR) is the distance from the receiver antenna to the satellite antenna including receiver and satellite clock offsets (and other biases, such as atmospheric delays) :*
PR=distance+c(receiver clock offset–satellite clock offset + other biases)*". It thus represents a raw estimate of the distance between the receiver and the space vehicle, independently of any propagation delay correction of the electromagnetic wave crossing the various atmospheric layers.

An example of a measurement, extracted from a RINEX file generated by a single-frequency (L1) U-Blox receiver with phase measurement, is

```
> 2017 12 22  5 57 46.0010000  0 12
G12  22028410.605   115760077.968        3307.683          46.000
G18  21024975.970   110486988.127        1088.977          45.000
G24  20360955.102   106997530.988        -437.104          49.000
J 1  37731461.503   198280150.949         713.158          45.000
J 2  37863385.498   198973438.883        -372.958          45.000
G15  21655567.700   113800790.795       -1526.538          48.000
G20  22301572.946   117195549.217       -2991.357          44.000
R16  21729491.824   116075061.937        3857.002          41.000
R15  19336042.668   103325965.774        -387.583          43.000
R 4  20461570.837   109570805.433       -1945.881          44.000
R14  21528858.057   114761049.343       -3182.688          38.000
R 5  19671117.697   105153436.303        1676.938          38.000
```

The first letter indicates the constellation, with G for GPS, R the Russian GLONASS, and J for the Japanese QZSS satellites in geosynchronous orbit between 32000 and 38000 km.

A quick overview of the second column of these measurements comforts us on their validity : the GPS constellation (space vehicles whose names start with "G") orbits at 20000 km above the surface of the Earth. The pseudo-ranges are hence included between about twenty thousand km, and this

altitude incremented by twice the Earth radius 2×6400 km (obviously a GPS satellite located on the opposite side of the Earth than the receiver cannot be seen, but that is a worst case). Here the satellite ranges are included between 20000 km and 22000 km for GPS, a bit less for GLONASS, in agreement with our expectations. Japan (these measurements were collected from Sendai, in Japan) has developed a location system based on geosynchronous satellite orbits at higher altitudes : here again measurements are in agreement with our expectations, since we find 37800 km. No European Galileo satellite (name starting with "E") is visible in this acquisition. In the 4th column, the Doppler shifts are also in the range of values described in the text. The 5th column hints at the signal power, and the 3rd column at a carrier phase information more difficult to assess.

Converting pseudo-ranges described in a RINEX file to a usable timestamp or position information is taken care of by the excellent opensource library `rtklib` (`www.rtklib.com`), whose usage is beyond the scope of this article.

Understanding the content of RINEX files is mandatory since it is thanks to these reference files that a user can improve, using post-processing, the estimate of the position of the receiver by including corrections such as the ionospheric delay – delay of the electromagnetic wave introduced by the varying density of electrons in the ionosphere. For this reason, users of higher grade receivers than those used by the general public with only the processed NMEA information (too late to process the raw data and improve the resolution) can download improved resolution satellite ephemeris (observed orbital parameter rather than predicted) as well as the various correction factors, thanks to the services of the IGS (*International GNSS Service*) which collects precise measurements in reference stations distributed on the surface of the Earth. The two types of RINEX files are the observations (extension ending with `o`) from the ground based receivers, allowing to correct observations made by a user on the field – these files will be of no interest to us here – and the navigations files (ending with `n`) which provide orbital parameters of the satellites, independently on any assumption on the ground based location (Fig. 2).

This second dataset, describing the satellite constellation orbital parameters and here acquired by processing the navigation messages transmitted by the satellites, will also be available in an improved resolution version on various sites dedicated to collecting and disseminating the products provided by IGS, listed at `https://kb.igs.org/hc/en-us/articles/202054393-IGS-FTP-Sites` – for example `ftp://cddis.gsfc.nasa.gov/gnss/products/` – to provide input information to generate the GPS spoofing signals.

A second parameter dictated by the physics of the constellation of spaceborne radiofrequency emitters is the power received on the ground, as defined by a link budget driven in particular by the Free Space Propagation Losses stating energy conservation – once again a physical principle that we should not violate during our spoofing attempts. The standards describing GPS does not state the power emitted from space but the power received on the ground : [11, p.14] tells us that GPS *must* provide at ground level a signal power of -160 dBW=-130 dBm on the L1 carrier at 1575.42 MHz. While we have described in [8] how this signal lies below thermal noise and hence cannot be visible on a spectrum analyzer unless a high gain such as found at a radiotelecope installation is used, this signal is raised by 30 dB during pulse compression achieved by cross-correlating the acquired signal with the (known) code associated with each satellite. The first important point of this analysis is that the **signal level remains excessively low at the receiver level** and any ground-based emitter will very easily overwhelm this power to blind the receiver. On the other hand, we shall see that some receivers check the receiver power level and **reject excessively powerful signals** which could not be broadcast from space.

GPS uses a 2-MHz bandwidth signal, so that any spoofing attack requires a source with such a bandwidth. We use Analog Devices' PlutoSDR, available for 85 euros from Mouser (since the supply has been exhausted at Farnell). This circuit is able to transmit up to 0 dBm (1 mW) and attenuate its output to lower this power.

For comparison, our emitter used to perform the attach can emit up to 1 mW (we check that a 0 dB attenuation matches a 0 dBm output power by measuring the level of a continuously emitted carrier), which we observe to drop to -30 dBm following spectrum spreading by phase modulation (Fig. 3). This observation is in agreement with the spectrum spreading over 1023 bits which lowers the peak power by $10\log_{10}(1023) = 30$ dB. This analysis valid for each satellite of the constellation whose signals add after propagation. The Free Space Propagation Loss ($FSPL$) are $FSPL = 20\log_{10}(d)+20\log_{10}(f)-147.55$, with the constant at the end of this equation given by $20\log_{10}(c/4/\pi)$ with $c = 3\cdot10^8$ m/s the velocity of an electromagnetic wave in vacuum. At $f =$ 1575.42 MHz, these losses amount to $20\log_{10}(d) +$ 36 dB. If we were to emit 0 dBm, then the losses needed to reach the -130 dBm of the standards are $FSPL = 130 = 20\log_{10}(d)+36$ dB, which would be reached at a distance of $d = 10^{(130-36)/20} = 50$ km. Since we actually emit 20 dB less (option `-A -20` of the GPS sentence generating software described below), the attack range is of the order of 5 km. Not considering the standards but the link budget in free space between a satellite emitting 25 W (`http://gpsinformation.net/main/gpspower.htm`) with an antenna gain of 13 dBi and the 182 dB free space propagation losses along the 20000 km spanning between the satellite and the surface of the Earth, the power seen at ground level would be -125 dBm. If we wish our signal to overwhelm the "real" signal by at least 3 dB, the the 8 dB difference with the previous calculation drop the range of the attack to 5 km$\times10^{(-8/20)}$=2 km, ensuring a low impact on the working environment of our tests : we have checked that, probably due to the poor dipole antenna powered by the PlutoSDR output with no balun [1], the GPS signal was stronger than our emitted signal at a range of about 50 meters from the emitter.

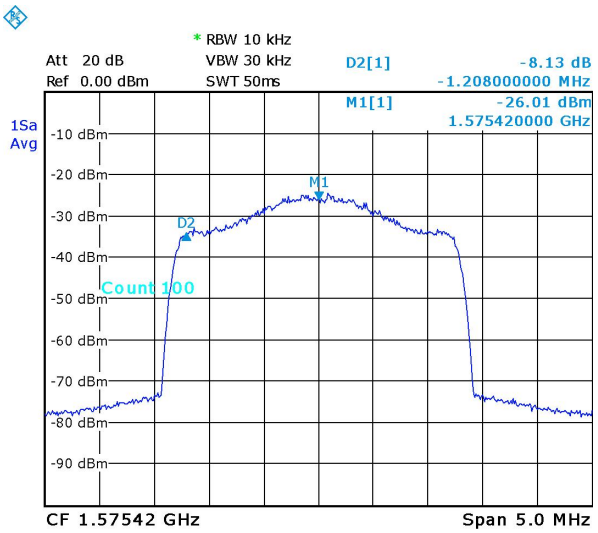


Figure 3: Spectrum of the signal emitted by a PlutoSDR tuned to a gain of 0 dB : the 1575.42 MHz carrier is spread over a spectrum spanning over ±1 MHz by the phase modulation along the pseudo-random sequence characterizing each satellite, inducing an about -30 dBm level in the band.

3 Software for spoofing attack deployment

Having selected a hardware platform meeting the carrier frequency (1575.42 MHz), bandwidth (2 MHz) and output power requirements, we must write the software synthesizing the signals. This work is not complex but requires carefully implementing all the steps : we rely on `github.com/Mictronics/pluto-gps-sim` to demonstrate the attack. This software is impressive in how compact it is yet since it implements all the steps, from reading the orbital parameter RINEX file to generating the navigation messages while going through the coordinates transforms needed for celestial mechanics, in about a thousand very readable lines of codes (and hence that can be updated to inject our own parameters in the transmitted messages).

The objective of the spoofing attack is to generate signals representative of those emitted by the satellite constellation. Considering that all satellites communicate through the same carrier frequency of 1575.42 MHz, the only challenge lies in generating the complex I/Q datastream as sum of the contributions of the various satellites, with the phase modulation of the code of each satellite Doppler shifted depending on the position of each satellite in the sky, and the navigation messages allowing to position the receiver on the surface of the Earth thanks to the delay introduced by the propagation of the signal from the satellite to ground as represented by each pseudo-range. In order not to be disturbed by the real satellites of the constellation emitting continuously, we must necessarily generate a spoofing signal including the satellites visible at the given time and place close to the location of the attack : were these requirements not met, the receiver would receive a mix of "real" signals and "false" signals and its chances of locking on the spoofed position are reduced. We have seen that at an altitude of 20000 km, satellite will run

1. A balun (*balanced-unbalanced*) is a transformer designed to convert the *un-balanced* signal (differentiating ground and signal) to a *balanced* signal to power the two symmetrical antenna wires.

through a full orbit in 12 h, so that exploiting a configuration valid a few hours prior to the attack remains relevant. The target location used during the attack must also not be too far from the receiver site itself so that the latter sees a constellation similar to the one included in the emitted messages. The list of the satellites and their orbital parameters as emitted in the navigation messages of the various satellites are published at `cddis.nasa.gov/Data_and_Derived_Products/GNSS/hourly_30second_data.html` with an hour resolution : this service is useful practically for correcting by post-processing GPS signals acquired from a unique receiver (correction of the ionospheric delay when no reference baseline station was available on site), as described earlier when introducing IGS.

The current GPS date day is fetched at `sopac.ucsd.edu/convertDate.shtml` : for example, July 30th 2018 is day 211 of the year, so the ephemeris are collected at `ftp://cddis.gsfc.nasa.gov/gnss/data/hourly/2018/211/`. Selecting the hour must obviously consider the offset between local time zone and universal time, namely +1 or +2 h in France. `cddis.nasa.gov/Data_and_Derived_Products/GNSS/broadcast_ephemeris_data.html#GPShourly` informs us that the filenames ending with `n` are those of interest to us (broadcast ephemeris) to know the orbital parameters of the space vehicles (SV) of the constellation : we select the file named `hour2110.18n.Z` (format `hourDDD0.YYn.Z` with `DDD` the day and `YY` the year)

```
./pluto-gps-sim -e hour2110.18n -A -20.0 -t 2018/07/30,10:00:00 -l 48.3621221,-4.8223307,100
Using static location mode.
Gain: -20.0dB
RINEX date = 30-JUL-18 23:30
Start time = 2018/07/30,10:00:00 (2012:122400)
PRN   Az    El     Range     Iono
04  110.0  80.4  20159594.4   1.7
05   33.5   8.9  24730267.9   4.4
09  320.7   5.1  25212521.1   4.5
16  302.7  51.7  21108967.6   2.0
20  144.1   7.2  25065494.6   6.2
21  133.7  64.8  21075459.6   1.9
23  292.6   5.3  25170257.9   4.5
25  120.9   6.6  25194957.8   6.4
26  292.6  82.7  20252112.2   1.7
27  256.8  22.9  23261110.3   3.3
29   64.7  31.3  22678543.7   3.1
31  193.6  33.0  22775272.7   3.0
```

The original tool, `gps-sdr-sim` from which `pluto-gps-sim` is derived, offers in addition to the static mode a dynamic mode, which requires however saving a rather large I/Q coefficient file (2.5 MS/s) computed prior to executing the attack, reducing its duration to a few minutes at most. This file is generated from a path defined in a NMEA formatted input file.

4 Demonstration : mobile phone and U-Blox receiver

The attack efficiency is first demonstrated on mobile phones, the geolocating tool most commonly used nowadays by the general public. Fig. 4 demonstrates the result of the attack on 3 phones : one of the smartphones has kept the coordinates of the local site obtained by exploiting the signals of the GPS constellation (Besançon at 47°N, 6°E), while the other two have been spoofed to an erroneous location arbitrarily selected South of France at 42.5°N, 2.3°E. Let us emphasize that in order to achieve such a result, we have deactivated any location assistance such as GSM or WiFi : this constraint is not restrictive since jamming is excessively simple to implement with respect to the complexity of jamming, and eliminating these location assistances is not a technical problem.

The same attack is completed successfully on Neo7M or NeoM8T U-Blox receivers. These receivers are interesting since in addition to being used on many drones including those sold by DJI, they provide raw information (pseudo-ranges) allowing to analyze in details the acquired signals, before processing to compute the position of the receiver. Indeed, the U-Blox Center sentence analysis tool provides many information on the received signals including *anti-spoofing* and *anti-jamming* characteristics. A first received power criterion rejects signal excessively powerful that could not have been broadcast from a satellite [12].

FIGURE 4 – Three smartphones are subject to the spoofing signal emitted by the PlutoSDR : one Samsung mobile phone (middle) and one Sony mobile phone (right) believe they are located South of France, while the left Samsung phone has stayed in Besançon.

Fig. 5 demonstrates on the receiver the impact of varying on purpose the frequency clocking the emitter. As the emitter clock is shifted by 5 ppm (200 Hz with respect to the nominal 40 MHz), the receiver keeps on providing positioning information despite the detection of the inconsistent measurements as indicated in the right columns named PR (Pseudo-Range), CP (Carrier Phase) and DO (Doppler Measurement) : the U-Blox receiver has detected the inconsistent values of the Doppler shift (red DO), but that does not prevent it, in its default configuration, from transmitting an erroneous position.

Attempting to spoof car GPS receivers using means similar to those used on mobile phones fails. We attribute this failure to spoof vehicles to the use of such inconsistency indicators on the received signal, namely an excessive power unrealistic from satellites in orbit, and inconsistent Doppler shifts. The first issue will be solved by tuning the output power, the second by using a frequency source more stable than the one originally provided with the PlutoSDR.

5 Demonstration : car GPS

This first spoofing experiment fails with some mobile phone models, but most worrying with all car GPS receivers. We attribute this failure to the offset between the PlutoSDR local oscillator frequency and its nominal frequency : even though the Rakon RXO3225M exhibits excellent performances for an oscillator based on a temperature compensated resonator, an offset of ± 25 ppm to the nominal frequency remains that a “real” GPS source would not generate. A rubidium clock as the one fitted in the space vehicles will exhibit a few ppb offset at worst, or at least a thousand fold better than this quartz oscillator.

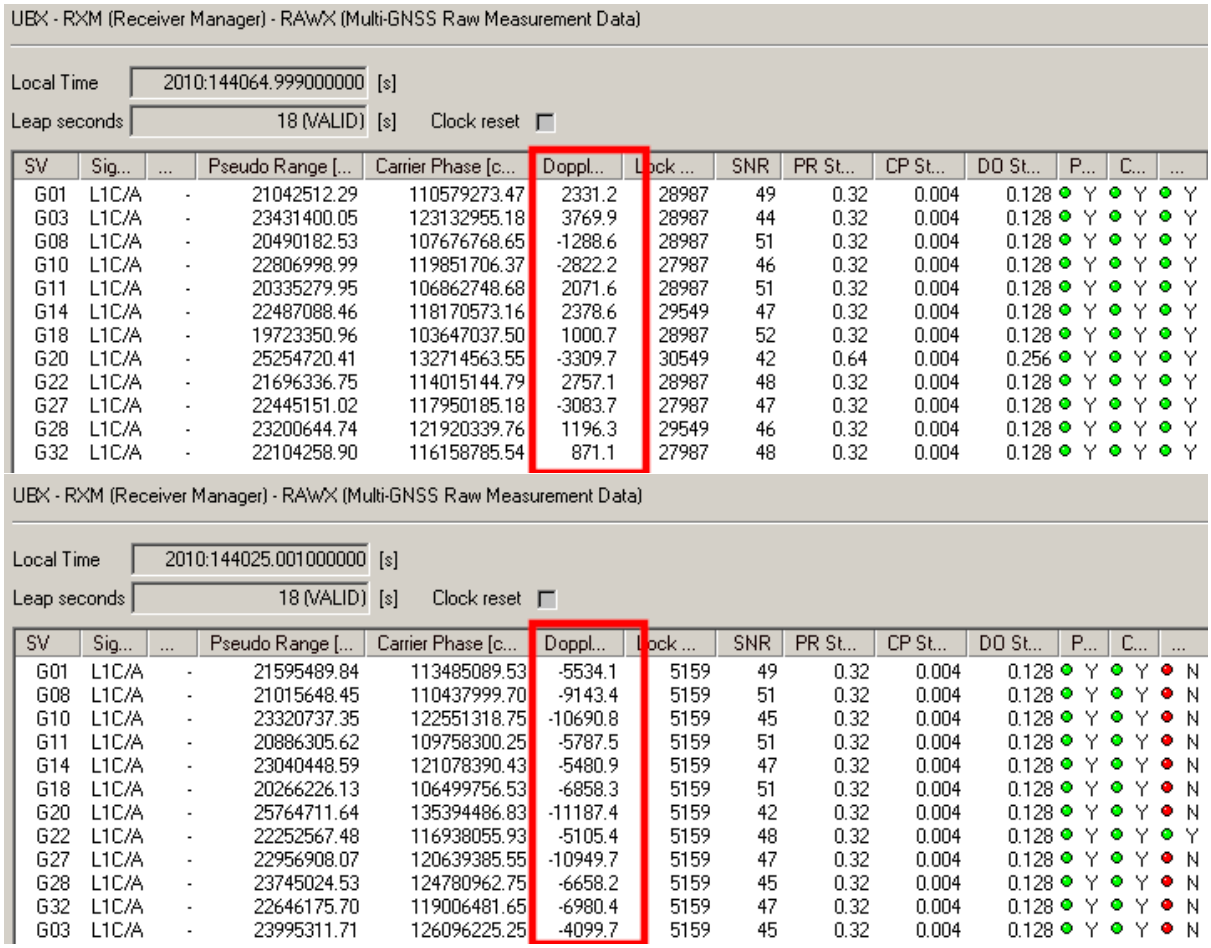
Our first solution to this local oscillator uncertainty is to exploit a synthesizer generating the 40 MHz output and controlled by a hydrogen maser known to be accurate. We shall remember, when running such an experiment, to cancel or remove the calibration coefficient included by Analog Devices in the software controlling the PlutoSDR. This can be done by logging in on the board using a terminal emulator or through `ssh` (login `root`, password `analog`), and executing :

```
echo 40000000 > /sys/bus/iio/devices/iio:device1/xo_correction
```

to claim that the frequency clocking the circuit is exactly 40 MHz. However, thus new definition of the local oscillator frequency is only active until the next reboot of the board. A long term sustainable solution consists in defining a new U-Boot non-volatile environment variable

```
fw_setenv xo_correction 40000000
```

Once this hardware modification is completed, and following the procedure described in the previous section, cars are also quickly spoofed, to bring cars parked in Besançon on the parking of École Nationale



UBX - RXM (Receiver Manager) - RAWX (Multi-GNSS Raw Measurement Data)

Local Time 2010:144064.999000000 [s]

Leap seconds 18 (VALID) [s] Clock reset

SV	Sig...	...	Pseudo Range [...	Carrier Phase [c...	Doppl...	Lock ...	SNR	PR St...	CP St...	DO St...	P...	C...	...
G01	L1C/A	-	21042512.29	110579273.47	2331.2	28987	49	0.32	0.004	0.128	Y	Y	Y
G03	L1C/A	-	23431400.05	123132955.18	3769.9	28987	44	0.32	0.004	0.128	Y	Y	Y
G08	L1C/A	-	20490182.53	107676768.65	-1288.6	28987	51	0.32	0.004	0.128	Y	Y	Y
G10	L1C/A	-	22806998.99	119851706.37	-2822.2	27987	46	0.32	0.004	0.128	Y	Y	Y
G11	L1C/A	-	20335279.95	106862748.68	2071.6	28987	51	0.32	0.004	0.128	Y	Y	Y
G14	L1C/A	-	22487088.46	118170573.16	2378.6	29549	47	0.32	0.004	0.128	Y	Y	Y
G18	L1C/A	-	19723350.96	103647037.50	1000.7	28987	52	0.32	0.004	0.128	Y	Y	Y
G20	L1C/A	-	25254720.41	132714563.55	-3309.7	30549	42	0.64	0.004	0.256	Y	Y	Y
G22	L1C/A	-	21696336.75	114015144.79	2757.1	28987	48	0.32	0.004	0.128	Y	Y	Y
G27	L1C/A	-	22445151.02	117950185.18	-3083.7	27987	47	0.32	0.004	0.128	Y	Y	Y
G28	L1C/A	-	23200644.74	121920339.76	1196.3	29549	46	0.32	0.004	0.128	Y	Y	Y
G32	L1C/A	-	22104258.90	116158785.54	871.1	27987	48	0.32	0.004	0.128	Y	Y	Y

UBX - RXM (Receiver Manager) - RAWX (Multi-GNSS Raw Measurement Data)

Local Time 2010:144025.001000000 [s]

Leap seconds 18 (VALID) [s] Clock reset

SV	Sig...	...	Pseudo Range [...	Carrier Phase [c...	Doppl...	Lock ...	SNR	PR St...	CP St...	DO St...	P...	C...	...
G01	L1C/A	-	21595489.84	113485089.53	-5534.1	5159	49	0.32	0.004	0.128	Y	Y	N
G08	L1C/A	-	21015648.45	110437999.70	-9143.4	5159	51	0.32	0.004	0.128	Y	Y	N
G10	L1C/A	-	23320737.35	122551318.75	-10690.8	5159	45	0.32	0.004	0.128	Y	Y	N
G11	L1C/A	-	20886305.62	109758300.25	-5787.5	5159	51	0.32	0.004	0.128	Y	Y	N
G14	L1C/A	-	23040448.59	121078390.43	-5480.9	5159	47	0.32	0.004	0.128	Y	Y	N
G18	L1C/A	-	20266226.13	106499756.53	-6858.3	5159	51	0.32	0.004	0.128	Y	Y	N
G20	L1C/A	-	25764711.64	135394486.83	-11187.4	5159	42	0.32	0.004	0.128	Y	Y	N
G22	L1C/A	-	22252567.48	116938055.93	-5105.4	5159	48	0.32	0.004	0.128	Y	Y	Y
G27	L1C/A	-	22956908.07	120639385.55	-10949.7	5159	47	0.32	0.004	0.128	Y	Y	N
G28	L1C/A	-	23745024.53	124780962.75	-6658.2	5159	45	0.32	0.004	0.128	Y	Y	N
G32	L1C/A	-	22646175.70	119006481.65	-6980.4	5159	47	0.32	0.004	0.128	Y	Y	N
G03	L1C/A	-	23995311.71	126096225.25	-4099.7	5159	45	0.32	0.004	0.128	Y	Y	N

FIGURE 5 – Impact of the local oscillator of the signal synthesizer on the Doppler shift observed by the U-Blox receiver. Here, a synthesizer controlled by a hydrogen maser clocks the PlutoSDR at its nominal frequency of 40 MHz (top) or at 40 MHz-200 Hz, or a 5 ppm shift. We observe that in the former case all Doppler shifts lie within the range authorized by celestial mechanics (±5 kHz), while in the second case the frequency shifts are inconsistent.

Supérieure de Mécanique et Microtechniques (ENSMM) to believe that their wheels are in the water off Brest (Fig. 6). Our hypothesis is the right one, the accuracy of the signal clocking the emitter is the cause of the failure of the initial attack on car GPS receivers.

Each Pluto is clocked by a 40 MHz oscillator whose exact frequency is calibrated and stored in an area of the memory which is not trivially accessible by the user. In our case, we wish to inform the PlutoSDR that it will from now on be clocked by a stable 10 MHz quartz oscillator.

Before launching the Linux kernel, U-Boot modifies the default value of the clock as defined in the *devicetree* by loading in memory the calibration value. Such a result is achieved by U-Boot by calling the script `adi_loadvals` which executes :

```
fdt set /clocks/clock@0 clock-frequency ${ad936x_ext_refclk}
```

The content of the `ad936x_ext_refclk` variable is loaded by reading the memory area dedicated to the calibration and its value is hence overwritten just before calling `adi_loadvals`, preventing the user from overloading this variable to replace it with a custom value.

This limitation is avoided as described at `ez.analog.com/university-program/f/q-a/77922/...`
`...will-it-be-possible-to-feed-in-a-reference-clock-to-the-adalm-pluto/295481#295481` by modifying the script and add a new variable which, if present, will be used instead of

FIGURE 6 – Top, right : setup in which the oscillator clocking the PlutoSDR is replaced either with the output of a frequency synthesizer controlled by a hydrogen maser (here not used), or an oven controlled quartz-crystal oscillator (OCXO). Bottom : two cars – Renault (left) and Mercedes (bottom, right) – located on the parking of ENSMM believe they have their wheels in the water off Brest.

`ad936x_ext_refclk`. Practically, the original script found at `github.com/analogdevicesinc/u-boot-xlnx/blob/pluto/include/configs/zynq-common.h#L271` becomes :

```
if test ! -n $"{ad936x_skip_ext_refclk}"; then if test -n $"{ad936x_custom_refclk}";
then fdt set /clocks/clock0 clock-frequency $"{ad936x_custom_refclk}"; elif
test -n $"{ad936x_ext_refclk}"; then fdt set /clocks/clock0 clock-frequency
$"{ad936x_ext_refclk}"; fi; fi;
```

We can now define the variable `ad936x_custom_refclk` with the value we wish to define as the clock frequency :

```
fw_setenv ad936x_custom_refclk "<10000000>"
```

Few readers will have access to a hydrogen maser, and the solution can hardly be brought anyway to the site of the attack. We overcome this deficiency by replacing the maser with a high quality quartz oscillator. While oscillators controlled by temperature compensated resonators (*Temperature Controlled Crystal Oscillator* – TCXO) exhibit frequency fluctuations of a few tens of ppm depending on environmental conditions, an *Oven Controlled Crystal Oscillator* – OCXO – exhibits fluctuations below a ppm. We have salvaged from a broken Hewlett Packard 5345A electronic counter and excellent HP10811 OCXO [2]. This oscillator exhibits relative frequency fluctuations of $5 \cdot 10^{-13}$ at a second integration time to rise to $5 \cdot 10^{-12}$ at 100 seconds and drift on the long term (Fig. 7). The quartz was tuned to better than 30 mHz of its nominal frequency by comparison with the hydrogen maser output. Here again, by controlling a frequency synthesizer with this source, the attack on the cars is successful, this time with a setup requiring only about one hundred mA at 24 V for heating and a few mA at 12 V for the oscillator itself. The PlutoSDR only needs to be tuned now to accept a 10 MHz source instead of the nominal

2. this oscillator is available for about 100 euros on eBay. Alternatively, a rubidium clock, available as a second-hand item at the same cost such as the Symmetricom X72, would be suited as well

40 MHz to get rid of the synthesizer (see box).

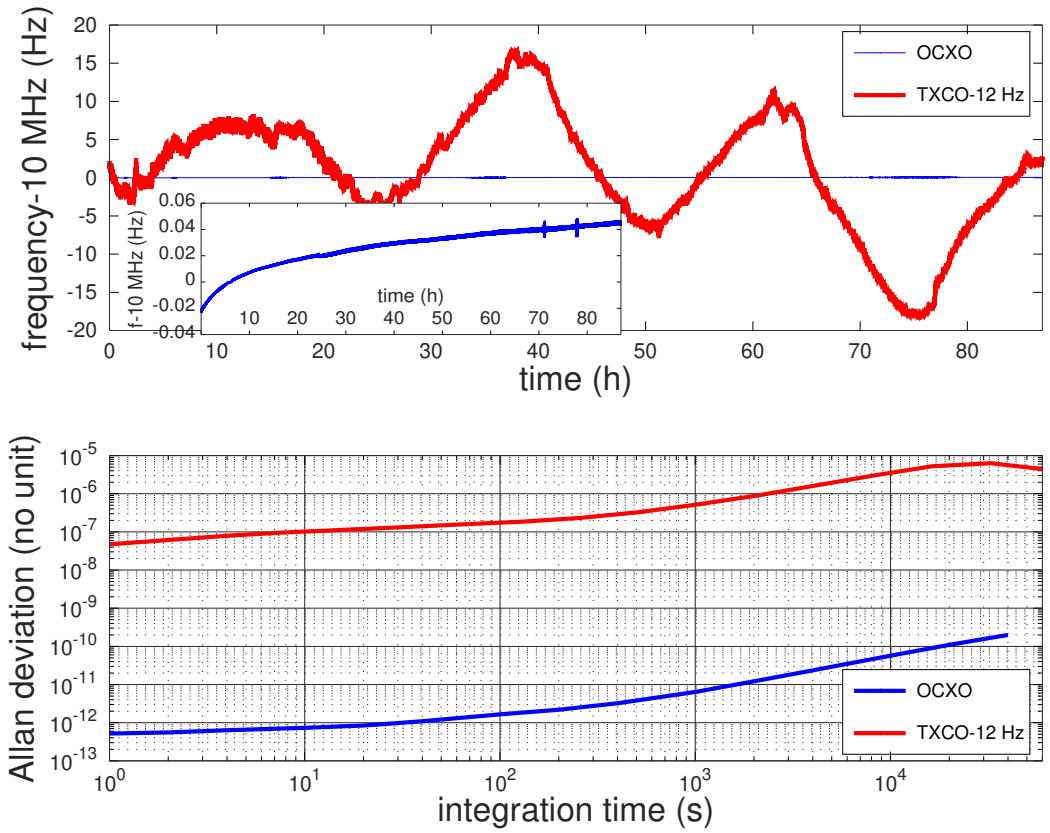


FIGURE 7 – Top : evolution over time of the frequency of the Rakon TCXO initially provided with the PlutoSDR (red) and a HP10811 OCXO. The inset exhibits a zoom on the OCXO measurement with its own dedicated scale. Bottom : Allan variance computed on the same dataset, illustrating the stability gain by 5 orders of magnitude when replacing the TCXO with the OCXO. All measurements are referred to a hydrogen maser : the TCXO is measured using an Agilent 53132A frequency counter, the OCXO is characterized using a Symmetricom TSC5110A analyzer.

6 Shifting time

A classical application of GPS for transferring time uses the 1 PPS signal output – 1 Pulse Per Second – which provides an accurate synchronization signal for controlling clocks on a common time base shared by the satellite constellation [13].

While frequency transfer is a relatively abstract concept for the general public (consider computer networks and Gb/s transfers – how to define the /s of Gb/s for two computers separated by several hundreds of kilometers ?), time transfer is a tangible concept ("I am late, I will not reach on time my meeting" – but how to make sure the expecting partner defines the meeting time on the same clock reference ?). Time and frequency transfer are two dual activities that do not have to tackle the same challenges. A frequency, or its integral the phase, describes one characteristics of a periodic signal : for example a sine wave at 10 MHz exhibits properties that repeat every 100 ns, and one cannot distinguish between one period and its neighbor 100 ns later. If the zero-crossing hardly varies over time with respect to a reference, the oscillator frequency can be controlled if needed, but no absolute timestamping is possible. On the opposite, time transfer requires a brief event ("now") – hence an intrinsically broadband signal, as opposed to frequency transfer which considers narrowband signals – and an absolute timestamping, and hence must not repeat quickly to leave enough time to provide all the information associated with a pulse (the date and time). The **1 PPS signal** (1 Pulse Per Second) [14, p.247] provides such an information : by definition, its rising edge is aligned with the information to be transmitted (beginning of the second), while its duration and hence position of its falling edge are not defined. In parallel to this rising edge, a digital information is usually transmitted to inform on the date and time associated with this edge. We are in the configuration of the speaking clock stating : "at the next stroke, the time will be XX h". GPS receivers are not the only source of 1 PPS information : White Rabbit, an implementation of PTP (Precision Time Protocol) driven by CERN, also provides its 1 PPS signal in parallel to frequency transfer with its 10 MHz oscillator. For demonstration purposes, the figure on the right exhibits an example of a one week-end measurement of the time delay *dt* between the 1 PPS provided by two independent White Rabbit links between ENSMM and the Besançon Observatory. The maximum fluctuations are about 200 ps, and the standard deviation is about 20 ps.

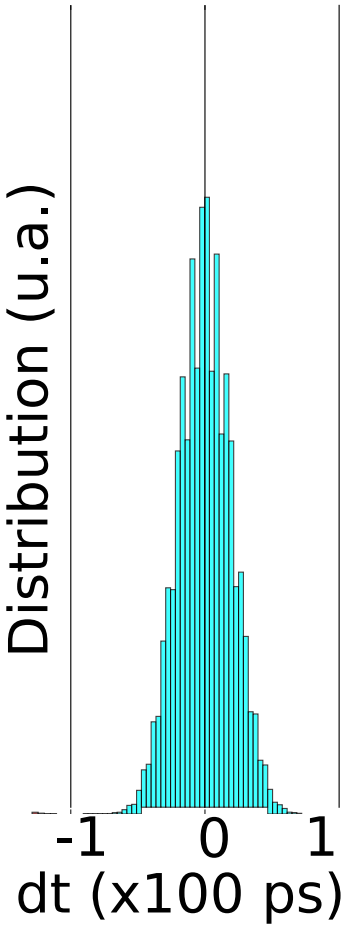


Figure 8: Delay distribution between two 1 PPS outputs from two White Rabbit links (chart drawn by É. Meyer, Obs. Besançon).

GPS is based on time transfer, from which the position on the ground of the receiver is computed by trilateration of the *pseudoranges*. The clock embedded onboard the satellite are not exact but drift. Rather than modify the behavior of these clocks to bring them to their nominal date, it is wise to leave the clocks drift in a deterministic way and inform the user of the offset between the time on the each clock embedded in each satellite and GPS time. This information, periodically updated, is transmitted in the navigation message broadcast by each satellite [9, p.57]. What happens if we offset this time with a known value, for example with 5 μs steps in the application that will follow ?

Performing this operation is simple in a software defined radio implementation : a navigation message parameter is updated consistently for all the satellites of the constellation, and hence time has been virtually translated. Since the pseudo-ranges are computed by `pluto-gps-sim` by including this timing offset for a given ground position, and since all clocks are anyway shifted by the same value (which hence compensate for during the trilateration), the receiver will not detect any position shift. Fig. 9 demonstrates this concept by exhibiting on the one hand the 1 PPS pulse representing GPS time for a Neo M8T U-Blox receiver, end on the other hand the position of this same receiver as provided by its NMEA sentences. We observe that the 1 PPS jumps with 5 μs steps as introduced on purpose every 2 minutes (remember that the 1 PPS of a single frequency GPS receiver typically fluctuates in the order of $\pm$100 ns), but that the position was by far not shifted by the 7 km that would have been expected by a time shift of 25 μs at the end of the 10 minutes experiment. A few position jumps are due to the time needed for the receiver feedback loop to converge when they get surprised by these sudden jumps

in time. We have also checked that the 1 PPS signal can be induced to drift linearly or quadratically by tuning not only the offset of the clock of each satellite but its first derivate (AF1 parameter) or second (AF2).

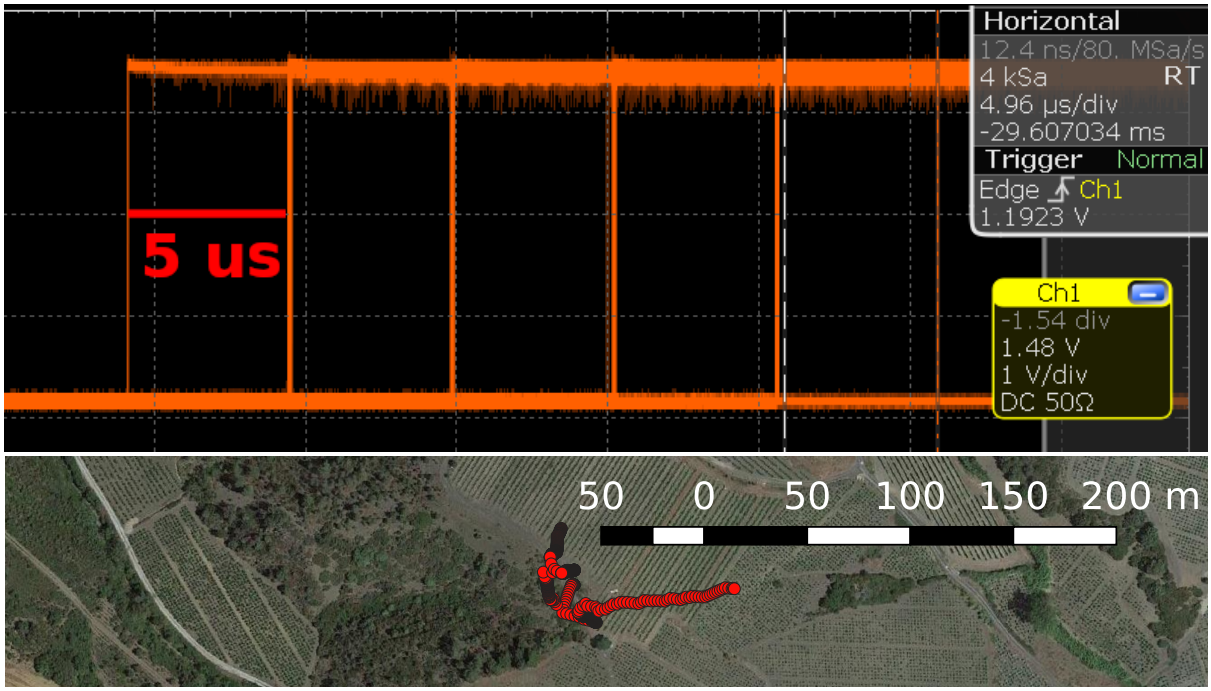


FIGURE 9 – Top : 1 PPS output of a U-Blox Neo M8T receiver indicating GPS time on which many oscillators (GPSDO) are controlled by trusting the signal coming from the satellite constellation, here induced to introduce errors with 5 μs steps. Bottom : negligible impact of the GPS time drift introduced by our attack on the receiver position estimate.

7 Palliative solutions

A recent special issue of Proc. IEEE summarized the state of the art attacks on satellite positioning systems. A first obvious solution is to consider data fusion from multiple sources in order to identify the origin of the inconsistency : adding to signals from multiple satellite constellations information emitted from ground such as WiFi or mobile phone networks (GSM, UMTS) reduces the risks but only delays the problem, since these additional sources can be jammed or spoofed (consider OpenBSC [15]).

A weakness of GNSS constellations lies in their very weak signal which is easily overwhelmed by ground-based emitters : a trend lies in positioning on the ground by using signals emitted by low-Earth orbiting satellites (e.g. Iridium NEXT), with signals much stronger at ground level than GPS and encrypted. Europe has unfortunately decided to give up on the deployment of a very low frequency (VLF) network of location emitters (eLORAN) which would provide a backup solution to attacks on GPS : the United States keep this network active and extend it to Japan and South Korea – both prone to jamming by their North Korean neighbor – while Saudi Arabia, China and Russia (Chayka) keep their VLF station network active to remain autonomous with respect to spaceborne positioning constellations. Spoofing the powerful VLF signals – 360 kW at 100 kHz for eLORAN – requires a setup with a much heavier infrastructure than a software defined radio emitter.

Finally, after describing the physical constraints defined by a satellite constellation (Doppler shift meeting Kepler's laws, distribution of the sources in space as defined by celestial mechanics), the ultimate solution seems to lie in the use of antenna networks (or alternatively a single moving antenna) to identify the direction of arrival of the signal and check for their consistency with the geometry of the constellation. Leading a distributed attack in which a multitude of time and frequency synchronized emitters generate signals able to reproduce the direction of arrival of each signal currently seems hardly accessible, and

such a solution seems the one favored by most articles in the review cited at the beginning of this section [16, 17, 18, 19, 20]. This approach fits well a fully software defined approach of the radio receiver, as demonstrated by the contribution of the authors of GNSS-SDR [21].

8 Conclusion

After reminding the reader with a few basic principles of GPS operation, from the physical layer to the software layer, we have demonstrated how easy it is nowadays to spoof GPS, even for such critical systems as car navigation systems. The objective is to make the reader aware of the dangers associated with a blind trust in satellite positioning systems, especially for critical infrastructures : in such cases, backup solutions to tackle jamming, or even when spoofing is detected through the inconsistency of the received signals (Doppler shift out of the range physically accessible, excessive received power) are needed. Finally, the multiple-antenna with measurement of the direction of arrival of the signals from the satellite constellation seems the most robust to prevent spoofing attacks.

Acknowledgements

This investigation has been motivated by the creation of the FASTLAB common laboratory between the FEMTO-ST institute, the Besançon Observatory and the company Gorgy Timing. The equipment acquired in the framework of the Labex FIRST-TF and Equipex OscillateurIMP provided the reference signals needed to qualify the various oscillators used in this presentation, although we insist on the ability of any enlightened amateur reader to reproduce these experiments. All literature references that are not freely available on the web have been fetched at the Library Genesis at `gen.lib.rus.ec`, an invaluable resource for our research activity.

Références

[1] T. Humphreys, *How to fool a GPS* at `www.ted.com/talks/todd_humphreys_how_to_fool_a_gps` (2012) and then `news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea`

[2] J.F. Zumberge & G. Gendt, *The demise of selective availability and implications for the international GPS service*, Physics and Chemistry of the Earth **26** (6–8), pp. 637–644 (2001) and `https://www.gps.gov/systems/gps/modernization/sa/`

[3] *Satellite-derived Time and Position : A Study of Critical Dependencies* `www.gov.uk/government/uploads/system/uploads/attachment_data/file/676675/satellite-derived-time-and-position-blackett-review.pdf`

[4] R.T. Ioannides, T. Pany, & G. Gibbons, *Known Vulnerabilities of Global Navigation Satellite Systems, Status, and Potential Mitigation Techniques*, Proc. IEEE **104** (6) 1174–1194 (June 2016)

[5] K. Zeng & *al.*, *All Your GPS Are Belong To Us : Towards Stealthy Manipulation of Road Navigation Systems*, 27th USENIX Security Symposium (2018), available at `people.cs.vt.edu/gangwang/sec18-gps.pdf`

[6] L. Huang & Q. Yang, *Low cost GPS simulator : GPS spoofing by SDR*, DEFCON 23 (2015) à `media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/DEFCON-23-Lin-Huang-Qing-Yang-GPS-Spoofing.pdf`

[7] D. Robinson *Using GPS Spoofing to control time*, DEFCON 25 (2017) `www.youtube.com/watch?v=isiuTNh5P34`

[8] J.-M Friedt, G. Cabodevila, *Exploitation de signaux des satellites GPS reçus par récepteur de télévision numérique terrestre DVB-T*, OpenSilicium **15** (2015) [in French]

[9] ESA, *GNSS data processing* (2013), available at `gssc.esa.int/navipedia/GNSS_Book/ESA_GNSS-Book_TM-23_Vol_I.pdf` and the list of exercises `gssc.esa.int/navipedia/GNSS_Book/ESA_GNSS-Book_TM-23_Vol_II.pdf`

[10] Standard *RINEX – The Receiver Independent Exchange Format, Version 3.03* available at `https://kb.igs.org/hc/en-us/article_attachments/202583897/RINEX_303.pdf`

[11] *Global Positioning System Standard Positioning Service Signal Specificiation* at `www.gps.gov/technical/ps/1995-SPS-signal-specification.pdf` (1995)

[12] A. Thiel & M. Ammann, *Anti-Jamming techniques in u-blox GPS receivers*, October 2009 at `www.u-blox.com/sites/default/files/products/documents/u-blox-AntiJamming_WhitePaper_%28GPS-X-09008%29.pdf`

[13] W. Lewandowski *& al.*, *Testing Motorola Oncore GPS Receiver and Temperature-Stabilized Antennas for Time Metrology*, Proc. 28th Annual Precise Time and Time Interval Systems and Applications Meeting (1996) at `https://tycho.usno.navy.mil/ptti/1996papers/Vol%2028_37.pdf`

[14] C. Audoin & B. Guinot, *The measurement of time – Time, frequency and the atomic clock*, Cambridge Univ. Press (2001)

[15] H. Welte, *OpenBSC network-side GSM stack*, SSTIC 2010, at `www.sstic.org/media/SSTIC2010/SSTIC-actes/Projet_OpenBSC/SSTIC2010-Slides-Projet_OpenBSC-welte.pdf`

[16] I.J. Gupta, I.M. Weiss, & A.W. Morrison, *Desired Features of Adaptive Antenna Arrays for GNSS Receivers*, Proc. IEEE **104** (6) 1195–1206 (June 2016)

[17] J.L. Volakis, A.J. O'Brien, & C.-C. Chen, *Small and Adaptive Antennas and Arrays for GNSS Applications*, Proc. IEEE **104** (6) 1221–1232 (June 2016)

[18] A. Broumandan, A. Jafarnia-Jahromi, S. Daneshmand & G. Lachapelle, *Overview of Spatial Processing Approaches for GNSS Structural Interference Detection and Mitigation*, Proc. IEEE **104** (6) 1246–1257 (June 2016)

[19] M. Cuntz, A. Konovaltsev & M. Meurer *Concepts, Development, and Validation of Multiantenna GNSS Receivers for Resilient Navigation*, Proc. IEEE **104** (6) 1288–1301 (June 2016)

[20] M.G. Amin, X. Wang, Y.D. Zhang, F. Ahmad, & E. Aboutanios, *Sparse Arrays and Sampling for Interference Mitigation and DOA Estimation in GNSS*, Proc. IEEE **104** (6) 1302–1317 (June 2016)

[21] C. Fernández-Prades, J. Arribas, & P. Closas, *Robust GNSS Receivers by Array Signal Processing : Theory and Implementation*, Proc. IEEE **104** (6) 1207–1220 (June 2016)